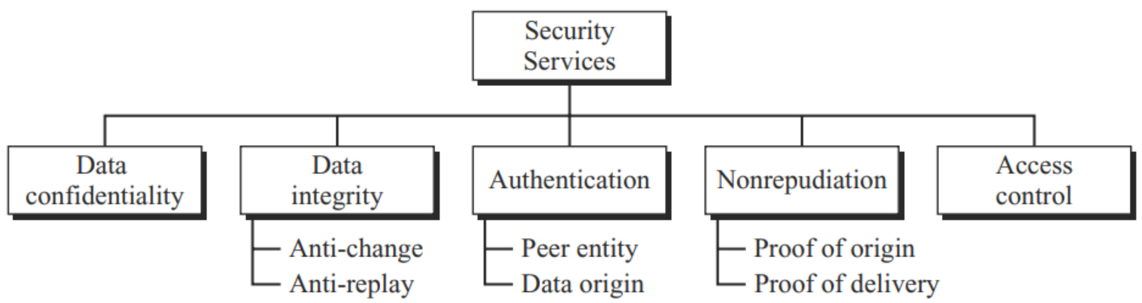
AC - Chapter 1

Created	@October 16, 2023 5:35 PM
Class	AC
Reviewed	☐

Security Goals

1. Confidentiality - assures that private or confidential information is not made available or disclosed to unauthorized individuals. A loss of confidentiality is the unauthorized disclosure of information.
2. Integrity - Assures that information (both stored and in transmitted packets) and programs are changed only in a specified and authorized manner. A loss of integrity is the unauthorized modification or destruction of information.
3. Availability - Assures that systems work promptly and service is not denied to authorized users. A loss of availability is the disruption of access to or use of information or an
information system.

Security Services



1. **Data Confidentiality**
Data confidentiality is designed to protect data from disclosure attack. The service as
defined by X.800 is very broad and encompasses confidentiality of the whole message
or part of a message and also protection against traffic analysis. That is, it is

designed to
prevent snooping and traffic analysis attack.

2. **Data Integrity**
   Data integrity is designed to protect data from modification, insertion, deletion, and
   replaying by an adversary. It may protect the whole message or part of the message.

3. **Authentication**
   This service provides the authentication of the party at the other end of the line. In
   connection-oriented communication, it provides authentication of the sender or receiver during the connection establishment (peer entity authentication). In connectionless communication, it authenticates the source of the data (data origin authentication).

4. **Nonrepudiation**
   Nonrepudiation service protects against repudiation by either the sender or the receiver
   of the data. In nonrepudiation with proof of the origin, the receiver of the data can later
   prove the identity of the sender if denied. In nonrepudiation with proof of delivery, the
   sender of data can later prove that data were delivered to the intended recipient.

5. **Access Control**
   Access control provides protection against unauthorized access to data. (The term
   access in this definition is very broad and can involve reading, writing, modifying, executing programs, and so on.)

Security Mechanisms

1. Encipherment
   Encipherment, hiding or covering data, can provide confidentiality. It can also be used
   to complement other mechanisms to provide other services.

2. Data Integrity
   The data integrity mechanism appends to the data a short check-value that has been

created by a specific process from the data itself. The receiver receives the data and the
check-value. He creates a new check-value from the received data and compares the
newly created check-value with the one received. If the two check-values are the same,
the integrity of data has been preserved.

3. Digital Signature
   A digital signature is a means by which the sender can electronically sign the data and
   the receiver can electronically verify the signature. The sender uses a process that
   involves showing that she owns a private key related to the public key that she has
   announced publicly. The receiver uses the sender's public key to prove that the message
   is indeed signed by the sender who claims to have sent the message.
4. Authentication Exchange
   In authentication exchange, two entities exchange some messages to prove their identity to each other. For example, one entity can prove that she knows a secret that only
   she is supposed to know.
5. Traffic Padding
   Traffic padding means inserting some bogus data into the data traffic to thwart the
   adversary's attempt to use the traffic analysis.
6. Routing Control
   Routing control means selecting and continuously changing different available routes
   between the sender and the receiver to prevent the opponent from eavesdropping on a
   particular route.
7. Notarization
   Notarization means selecting a third trusted party to control the communication between two entities. This can be done, for example, to prevent repudiation. The receiver can involve a trusted party to store the sender request in order to

prevent the
sender from later denying that she has made such a request.

8. Access Control
   Access control uses methods to prove that a user has access right to the data or resources owned by a system. Examples of proofs are passwords and PINs.

Relation between Services and Mechanisms

Security Service	*Security Mechanism*
Data confidentiality	Encipherment and routing control
Data integrity	Encipherment, digital signature, data integrity
Authentication	Encipherment, digital signature, authentication exchanges
Nonrepudiation	Digital signature, data integrity, and notarization
Access control	Access control mechanism

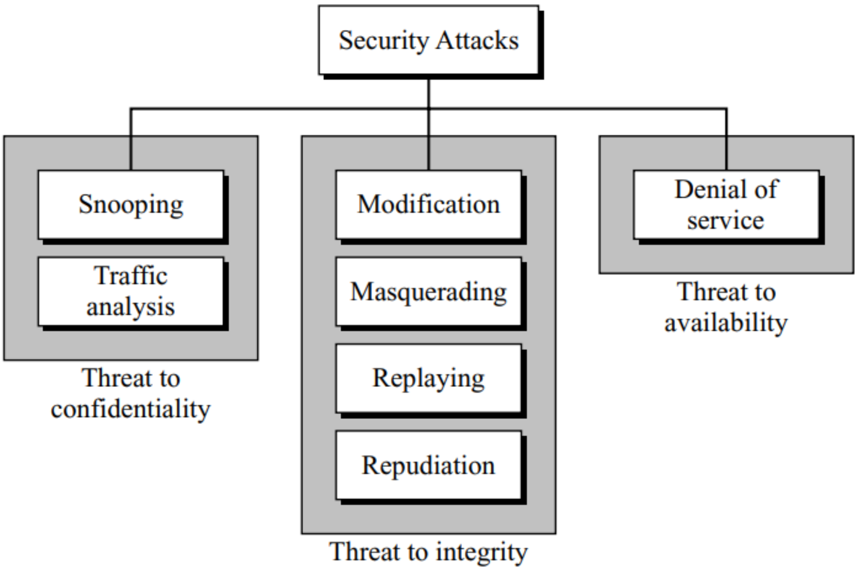
Threats

1. Innocent persons
2. Script kiddies
3. Hackers/crackers
4. Insiders
5. Nations
6. Natural disasters
7. Malicious software

Vulnerabilities

1. Lack of strong passwords
2. Lack of malware removal tools
3. Poor Access Control
4. Unpatched software
5. Device misconfiguration

Attacks



Confidentiality

1. **Snooping**
   Snooping refers to unauthorized access to or interception of data. For example, a file
   transferred through the Internet may contain confidential information. An unauthorized
   entity may intercept the transmission and use the contents for her own benefit. To prevent
   snooping, the data can be made non-intelligible to the intercepter by using encipherment
   techniques discussed in this book.
2. **Traffic Analysis**
   Although encipherment of data may make it non-intelligible for the intercepter, she can
   obtain some other type information by monitoring online traffic. For example, she can
   find the electronic address (such as the e-mail address) of the sender or the receiver. She
   can collect pairs of requests and responses to help her guess the nature of transaction.

Integrity

1. **Modification**
   After intercepting or accessing information, the attacker modifies the information

to
make it beneficial to herself. For example, a customer sends a message to a bank to do
some transaction. The attacker intercepts the message and changes the type of transaction to benefit herself. Note that sometimes the attacker simply deletes or delays the
message to harm the system or to benefit from it.

2. **Masquerading**
   Masquerading, or spoofing, happens when the attacker impersonates somebody else.
   For example, an attacker might steal the bank card and PIN of a bank customer and pretend that she is that customer. Sometimes the attacker pretends instead to be the
   receiver entity. For example, a user tries to contact a bank, but another site pretends that
   it is the bank and obtains some information from the user.
3. **Replaying**
   Replaying is another attack. The attacker obtains a copy of a message sent by a user and
   later tries to replay it. For example, a person sends a request to her bank to ask for payment to the attacker, who has done a job for her. The attacker intercepts the message
   and sends it again to receive another payment from the bank.
4. **Repudiation**
   This type of attack is different from others because it is performed by one of the two
   parties in the communication: the sender or the receiver. The sender of the message
   might later deny that she has sent the message; the receiver of the message might later
   deny that he has received the message.

Availability

Denial of service (DoS) is a very common attack. It may slow down or totally interrupt the service of a system. She might send so many bogus requests to a server that the server crashes because of the heavy load. The attacker might intercept and delete a server’s response to a client, making the client to believe that the server is not responding. The attacker may also intercept requests from the clients, causing the clients to send requests many times and overload the system.

1. Denial of Service
2. SYN-Flooding
3. IP Spoofing
4. Ping of death

AC - Chapter 2

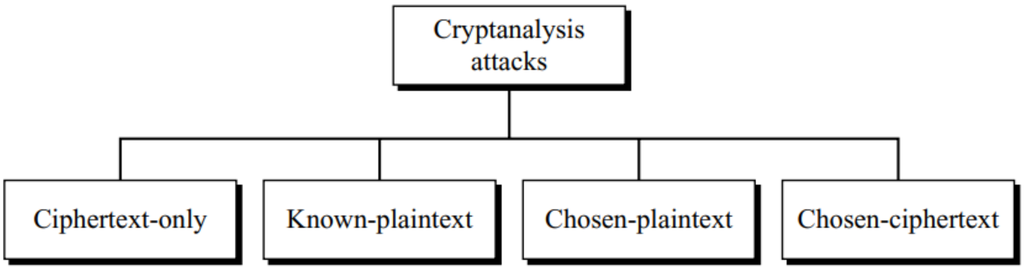
Created	@October 16, 2023 6:47 PM
Class	AC
Reviewed	☐

Symmetric Key Encryption

Symmetric-key encipherment uses a single key (the key itself may be a
set of values) for both encryption and decryption. In addition, the encryption and decryption algorithms are inverses of each other. If P is the plaintext, C is the ciphertext, and K is
the key, the encryption algorithm Ek(x) creates the ciphertext from the plaintext; the decryption algorithm Dk(x) creates the plaintext from the ciphertext. We assume that Ek(x) and Dk(x) are inverses of each other: they cancel the effect of each other if they are
applied one after the other on the same input

Cryptanalysis

Cryptanalysis is the science and art of breaking those codes.



Categories of Traditional Ciphers

1. Substitution Cipher
2. Transposition Cipher

Substitution Cipher

A substitution cipher replaces one symbol with another.

1. Monoalphabetic cipher:

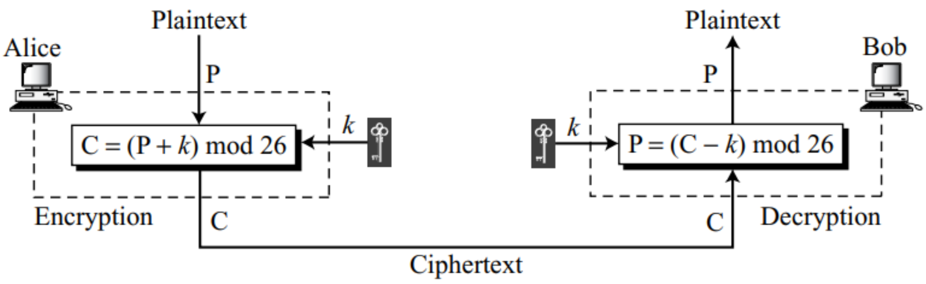
   In monoalphabetic substitution, a character (or a symbol) in the plaintext is always
   changed to the same character (or symbol) in the ciphertext regardless of its position in
   the text.

2. Polyalphabetic cipher:

   In polyalphabetic substitution, each occurrence of a character may have a different
   substitute. The relationship between a character in the plaintext to a character in the
   ciphertext is one-to-many.

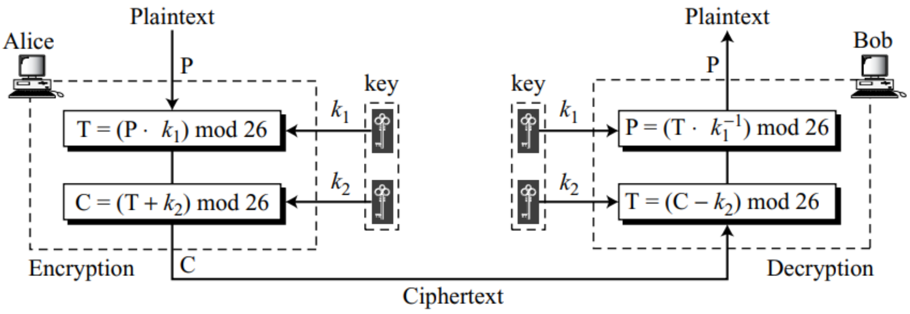
Substitution ciphers

1. Caesar Cipher - The simplest monoalphabetic cipher is the additive cipher. This cipher is sometimes called a shift cipher and sometimes a Caesar cipher.



2. Vigenere Cipher - A Vigenere cipher uses a different strategy to create the key stream. The key stream is a repetition of an initial secret key stream of length m, where we have $1 \leq m \leq 26$.

3. Affine Cipher - Uses both additive and multiplicative techniques, a combination of both ciphers with a pair of keys. The first key is used with the multiplicative cipher; the second key is used with the additive cipher. $C = ((P \times k1) + k2) \bmod 26$ and $P = ((C - k2) \times k1^{(-1)}) \bmod 26$.



In the affine cipher, the relationship between the plaintext P and the ciphertext C is

$$C = (P \times k_1 + k_2) \bmod 26 \qquad P = ((C - k_2) \times k_1^{-1}) \bmod 26$$

where k_1^{-1} is the multiplicative inverse of k_1 and $-k_2$ is the additive inverse of k_2

4. Playfair cipher - The secret key in this cipher is made of 25 alphabet letters arranged in a 5 × 5 matrix (letters I and J are considered the same when encrypting).

The cipher uses three rules for encryption:

a. If the two letters in a pair are located in the same row of the secret key, the corresponding encrypted character for each letter is the next letter to the right in the
same row (with wrapping to the beginning of the row if the plaintext letter is the last character in the row).

b. If the two letters in a pair are located in the same column of the secret key, the corresponding encrypted character for each letter is the letter beneath it in the same
column (with wrapping to the beginning of the column if the plaintext letter is the last character in the column).

c. If the two letters in a pair are not in the same row or column of the secret, the corresponding encrypted character for each letter is a letter that is in its own row but
in the same column as the other letter.

Secret Key =

L	G	D	B	A
Q	M	H	E	C
U	R	N	I/J	F
X	V	S	O	K
Z	Y	W	T	P

Hill Cipher

Unlike the other polyalphabetic ciphers we have already discussed, the
plaintext is divided into equal-size blocks. The blocks are encrypted one at a time in such a way that each character in the block contributes to the encryption of other characters in the block. In a Hill cipher, the key is a square matrix of size m × m in which m is the size
of the block.

$$\overset{\mathbf{C}}{\begin{bmatrix} 14 & 07 & 10 & 13 \\ 08 & 07 & 06 & 11 \\ 11 & 08 & 18 & 18 \end{bmatrix}} = \overset{\mathbf{P}}{\begin{bmatrix} 02 & 14 & 03 & 04 \\ 08 & 18 & 17 & 04 \\ 00 & 03 & 24 & 25 \end{bmatrix}} \overset{\mathbf{K}}{\begin{bmatrix} 09 & 07 & 11 & 13 \\ 04 & 07 & 05 & 06 \\ 02 & 21 & 14 & 09 \\ 03 & 23 & 21 & 08 \end{bmatrix}}$$

a. Encryption

$$\overset{\mathbf{P}}{\begin{bmatrix} 02 & 14 & 03 & 04 \\ 08 & 18 & 17 & 04 \\ 00 & 03 & 24 & 25 \end{bmatrix}} = \overset{\mathbf{C}}{\begin{bmatrix} 14 & 07 & 10 & 13 \\ 08 & 07 & 06 & 11 \\ 11 & 08 & 18 & 18 \end{bmatrix}} \overset{\mathbf{K}^{-1}}{\begin{bmatrix} 02 & 15 & 22 & 03 \\ 15 & 00 & 19 & 03 \\ 09 & 09 & 03 & 11 \\ 17 & 00 & 04 & 07 \end{bmatrix}}$$

b. Decryption

Transposition Cipher

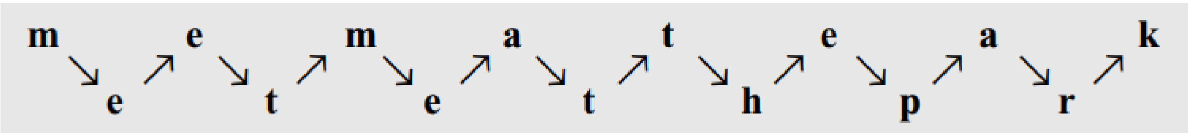
Columnar Transposition

Alice and Bob can agree on the number of columns and use the second method. Alice writes the
same plaintext, row by row, in a table of four columns.

m	e	e	t
m	e	a	t
t	h	e	p
a	r	k	

Rail fence Cipher

In this cipher, the plaintext is arranged in two lines as a zigzag pattern (which means column by column); the ciphertext is created reading the pattern row by row.



Ciphertext “MEMATEAKETETHPR”

AC - Chapter 3

Created	@October 16, 2023 7:29 PM
Class	AC
Reviewed	☐

Stream and Block Ciphers

Stream Cipher -

In a modern stream cipher, each r-bit word in the plaintext stream is enciphered using an r-bit word in the key stream to create the corresponding r-bit
word in the ciphertext stream.

Stream ciphers are faster than block ciphers. The hardware implementation of a stream cipher is also easier. When we need to encrypt binary streams and transmit them
at a constant rate, a stream cipher is the better choice to use. Stream ciphers are also
more immune to the corruption of bits during transmission.

Block Cipher -

A symmetric-key modern block cipher encrypts an n-bit block of plaintext or decrypts an n-bit block of ciphertext. The encryption or decryption algorithm uses a k-bit key. The
decryption algorithm must be the inverse of the encryption algorithm, and both operations
must use the same secret key.

Feistel Cipher

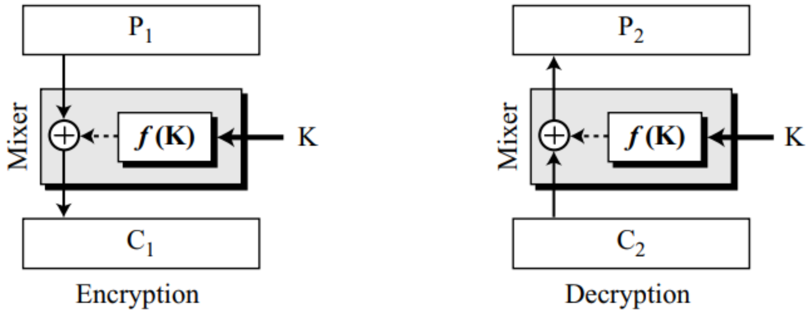
A Feistel cipher can have three types of components: self-invertible, invertible, and noninvertible. A Feistel cipher combines all noninvertible elements in a unit and uses the same unit in the encryption and decryption algorithms.

Useful properties of Ex-OR Operation

1. Does not lose info
2. Reversible

3. Induces randomness in algo
4. Cancels effect of encryption during decryption

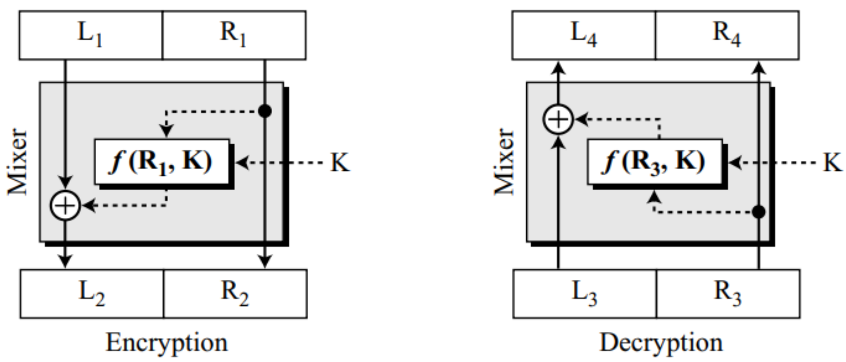
First Draft



In the encryption, a noninvertible function, f(K), accepts the key as the input. The output of this component is exclusive-ored with the plaintext. The result becomes the ciphertext. We call the combination of the function and the exclusive-or operation the mixer (for lack of another name).

Second Draft

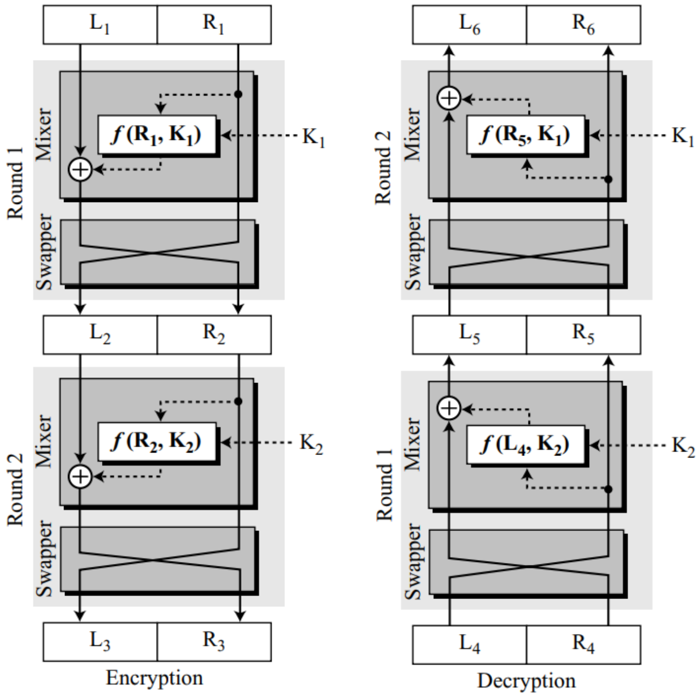
The key can be used as the second input to the function. In this way, our function can be a complex element with some keyless elements and some keyed elements. To achieve this
goal, divide the plaintext and the ciphertext into two equal-length blocks, left and right.
We call the left block L and the right block R. Let the right block be the input to the function, and let the left block be exclusive-ored with the function output.



$$R_4 = R_3 = R_2 = R_1$$
$$L_4 = L_3 \oplus f(R_3, K) = L_2 \oplus f(R_2, K) = L_1 \oplus f(R_1, K) \oplus f(R_1, K) = L_1$$

Final Draft

In second draft, the right half of the plaintext never changes. Eve can immediately find the right half of the plaintext by intercepting the ciphertext and extracting the right half of it. The design needs more improvement. First, increase the number of rounds. Second, add a new element to
each round: a swapper. The effect of the swapper in the encryption round is canceled by the effect of the swapper in the decryption round. However, it allows us to swap the left and right halves in each round.

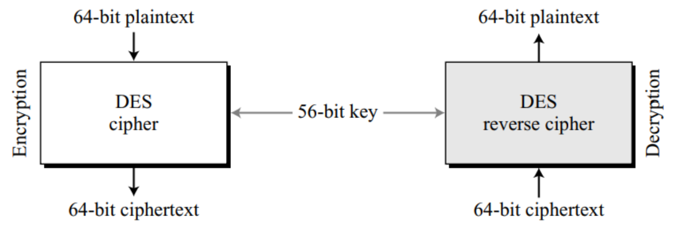


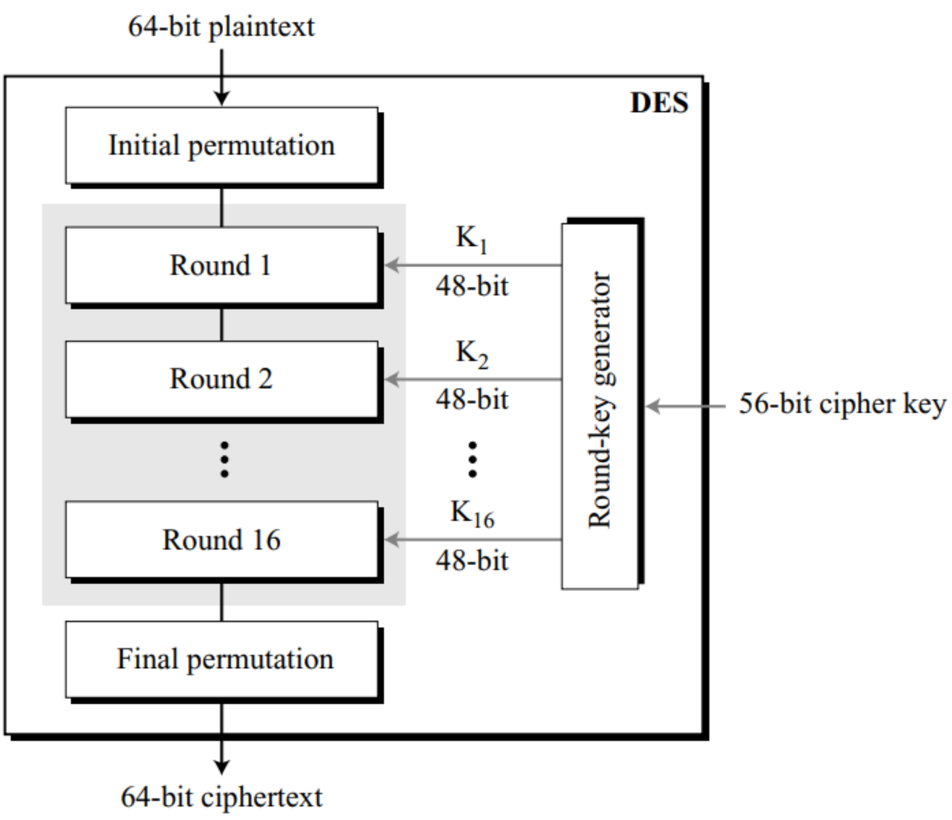
Note that there are two round keys, K1 and K2. The keys are used in reverse order
in the encryption and decryption.
Because the two mixers are inverses of each other, and the swappers are inverses of
each other, it should be clear that the encryption and decryption ciphers are inverses
of each other.

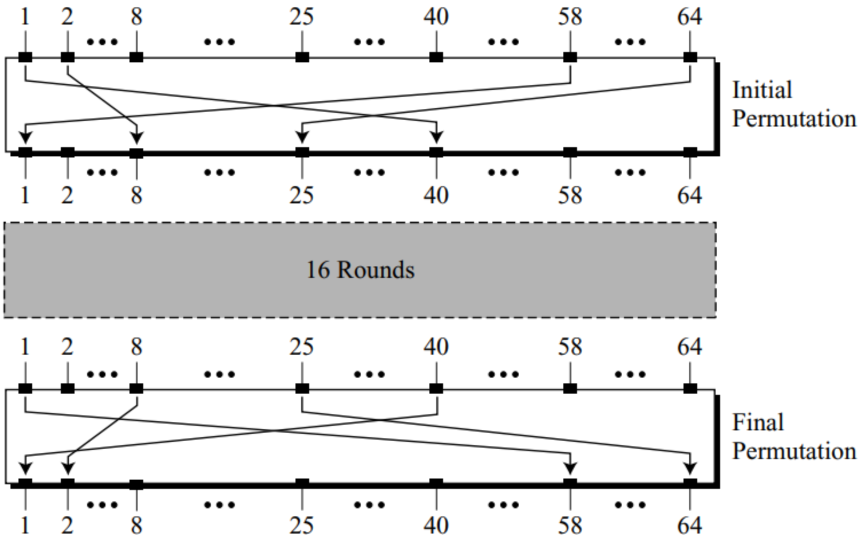
$$L_5 = R_4 \oplus f(L_4, K_2) = R_3 \oplus f(R_2, K_2) = L_2 \oplus f(R_2, K_2) \oplus f(R_2, K_2) = L_2$$
$$R_5 = L_4 = L_3 = R_2$$

DES (Data Encryption Standard)

The Data Encryption Standard (DES) is a symmetric-key block cipher published by
the National Institute of Standards and Technology (NIST), in March 1975.

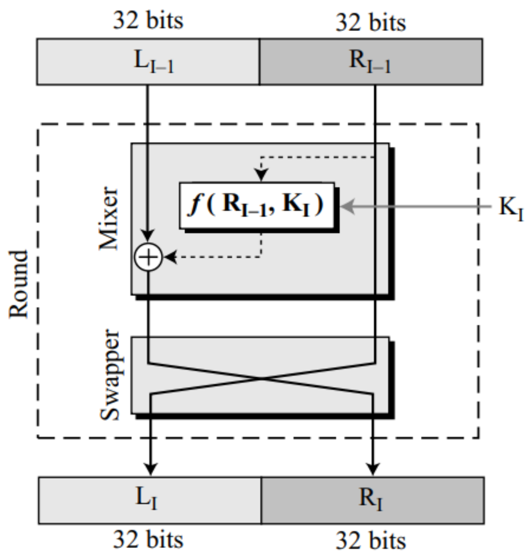




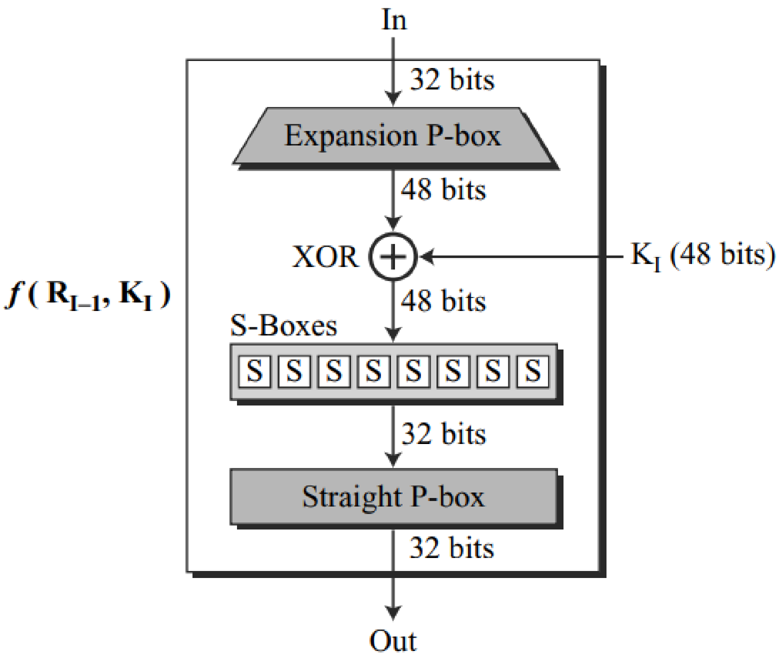


The initial and final permutations are straight P-boxes that are inverses of each other.
They have no cryptography significance in DES.

Figure 6.4 *A round in DES (encryption site)*



DES Function



1. Expansion P-box: Since RI−1 is a 32-bit input and KI is a 48-bit key, we first need to
   expand RI−1 to 48 bits. RI−1 is divided into 8 4-bit sections. Each 4-bit section is then
   expanded to 6 bits. This expansion permutation follows a predetermined rule. For each
   section, input bits 1, 2, 3, and 4 are copied to output bits 2, 3, 4, and 5, respectively. Output bit 1 comes from bit 4 of the previous section; output bit 6 comes from bit 1 of the
   next section.

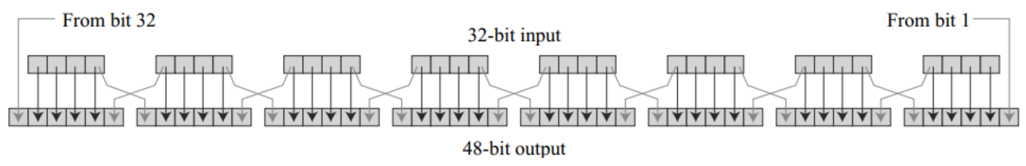


Table 6.2 *Expansion P-box table*

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	01

2. Whitener (XOR): After the expansion permutation, DES uses the XOR operation on
   the expanded right section and the round key. Note that both the right section and the
   key are 48-bits in length. Also note that the round key is used only in this operation.

3. S-Boxes: The S-boxes do the real mixing (confusion). DES uses 8 S-boxes, each with
   a 6-bit input and a 4-bit output.

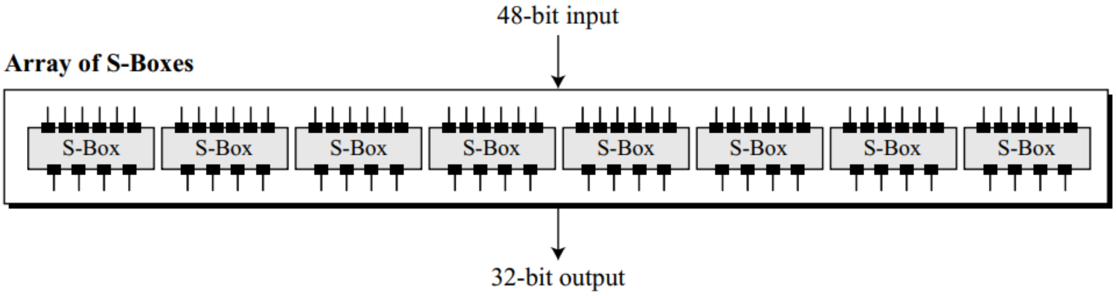
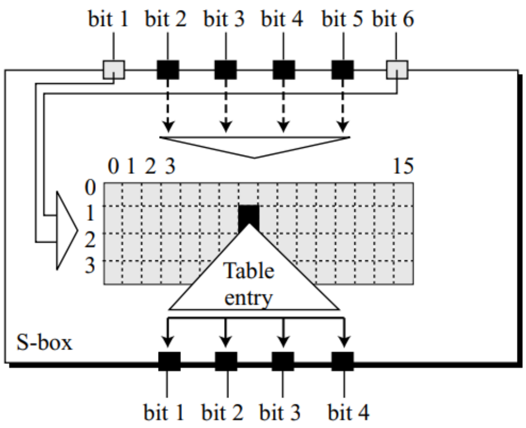


Figure 6.8 *S-box rule*

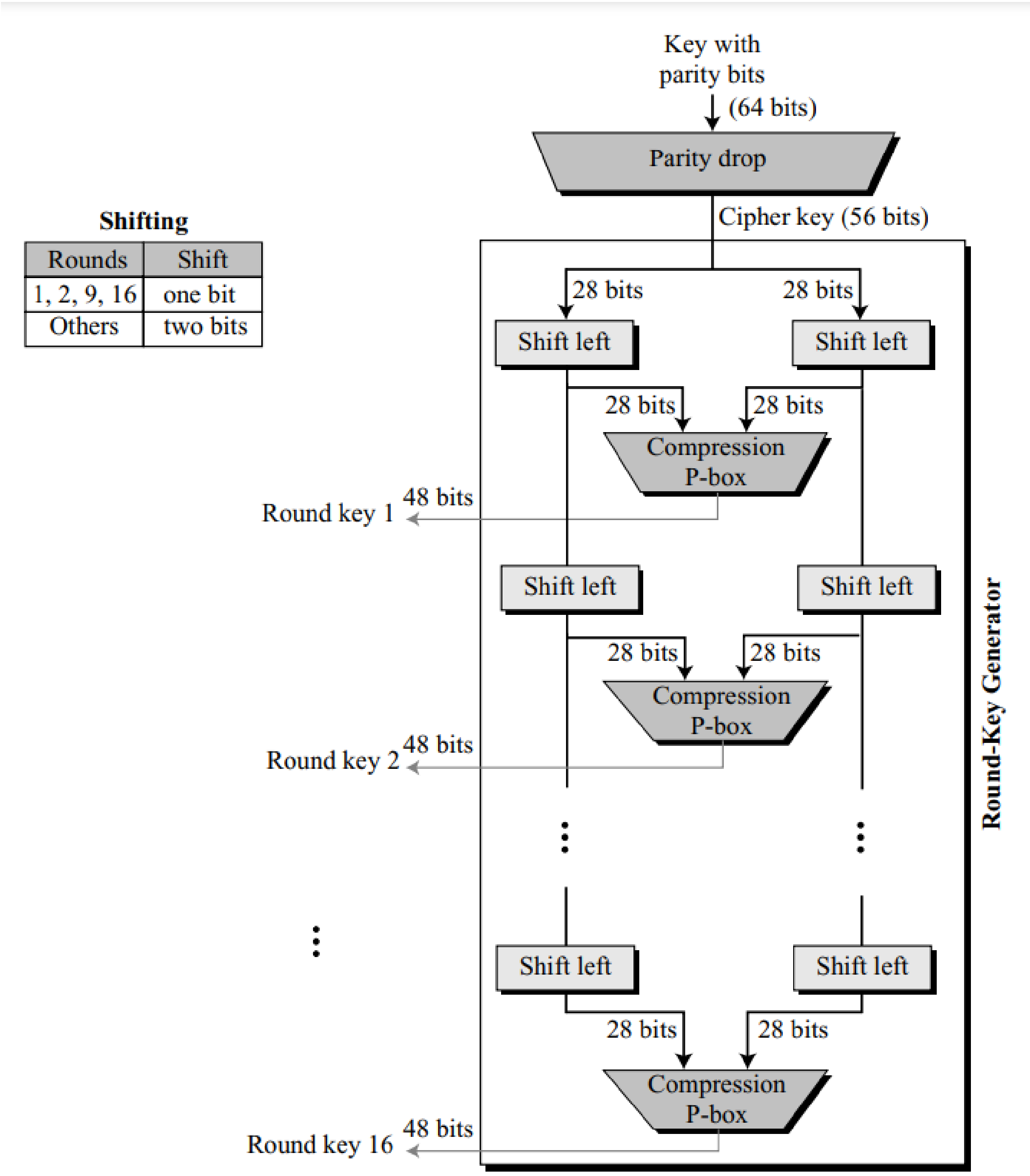


These are 8 separate tables for 8 separate S-boxes.

4. Straight Permutation: The last operation in the DES function is a straight permutation
with a 32-bit input and a 32-bit output.

Key Generation

The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key. However,
the cipher key is normally given as a 64-bit key in which 8 extra bits are the parity bits,
which are dropped before the actual key-generation process.



1. Shift Left
   After the straight permutation, the key is divided into two 28-bit parts. Each part is
   shifted left (circular shift) one or two bits
2. Compression Permutation
   The compression permutation (P-box) changes the 58 bits to 48 bits, which are used as
   a key for a round.

Properties

1. Avalanche Effect - Avalanche effect means a small change in the plaintext (or key) should create a significant change in the ciphertext. DES has been

proved to be strong with regard to this property.

2. Completeness - Completeness effect means that each bit of the ciphertext needs to depend on many bits on the plaintext. The diffusion and confusion produced by P-boxes and S-boxes in DES, show a very strong completeness effect.

Limitations of DES

1. Trapdoor possibility
2. Weak Cipher Keys
3. Attacks possible as follows

Security of DES

1. Brute-Force Attack: It is clear that DES can be broken using 255 encryptions. However, today most applications use either 3DES with two keys (key
   size of 112) or 3DES with three keys (key size of 168). These two multiple-DES versions make DES resistant to brute-force attacks.

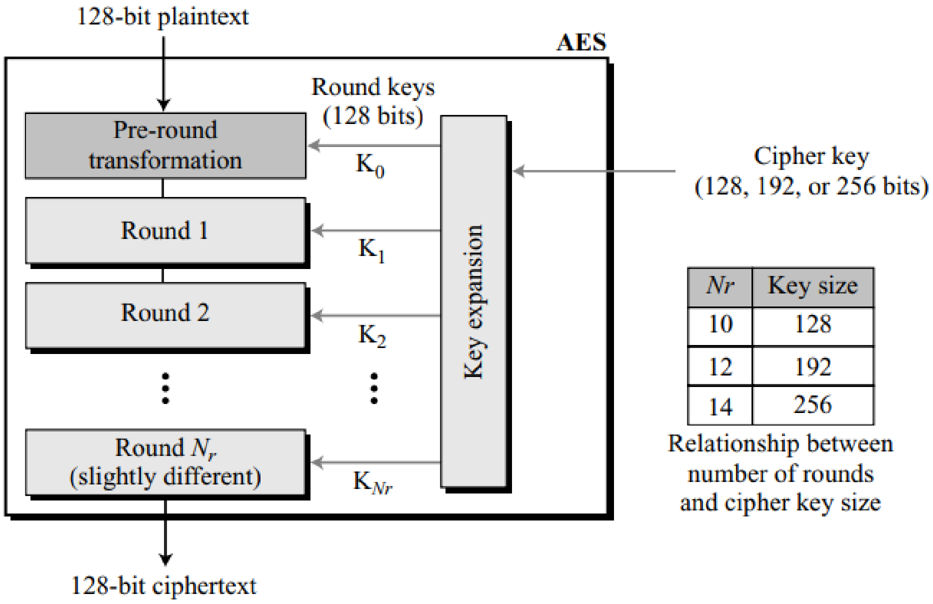
2. Differential Cryptanalysis: Today, it has been shown that DES can be broken using differential cryptanalysis
   if we have 247 chosen plaintexts or 255 known plaintexts. Although this looks more efficient than a brute-force attack, finding 247 chosen plaintexts or 255 know plaintexts is
   impractical. Therefore, we can say that DES is resistant to differential cryptanalysis.

3. Linear Cryptanalysis: S-boxes are not very resistant to linear cryptanalysis. It has been shown that DES can be broken using 243 pairs of known plaintexts.

AES (Advanced Encryption Standards)

AES was published by NIST in December 2001. AES is a non-Feistel cipher that encrypts and decrypts a data block of 128 bits. It uses
10, 12, or 14 rounds. The key size, which can be 128, 192, or 256 bits, depends on the
number of rounds.

Figure 7.1 *General design of AES encryption cipher*

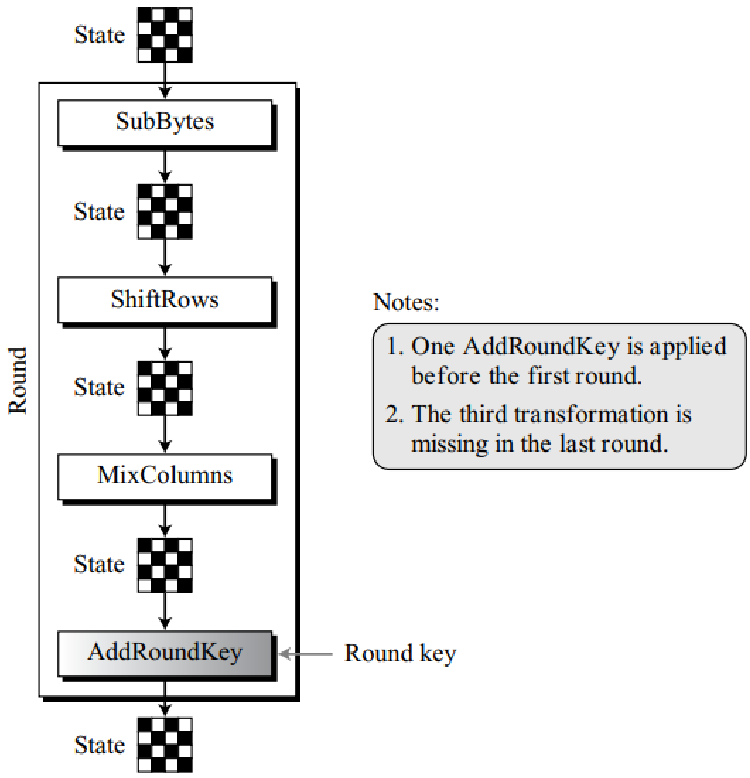


The number of round keys generated by the key-expansion algorithm is always one

more than the number of rounds. In other words, we have

Number of round keys = Nr + 1

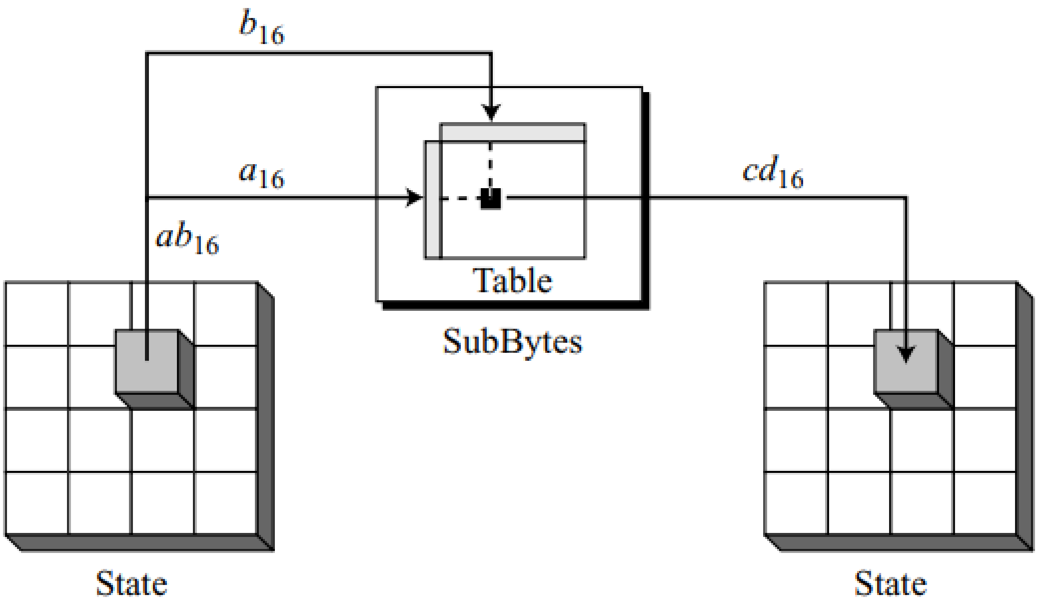
Figure 7.5 *Structure of each round at the encryption site*



SubBytes

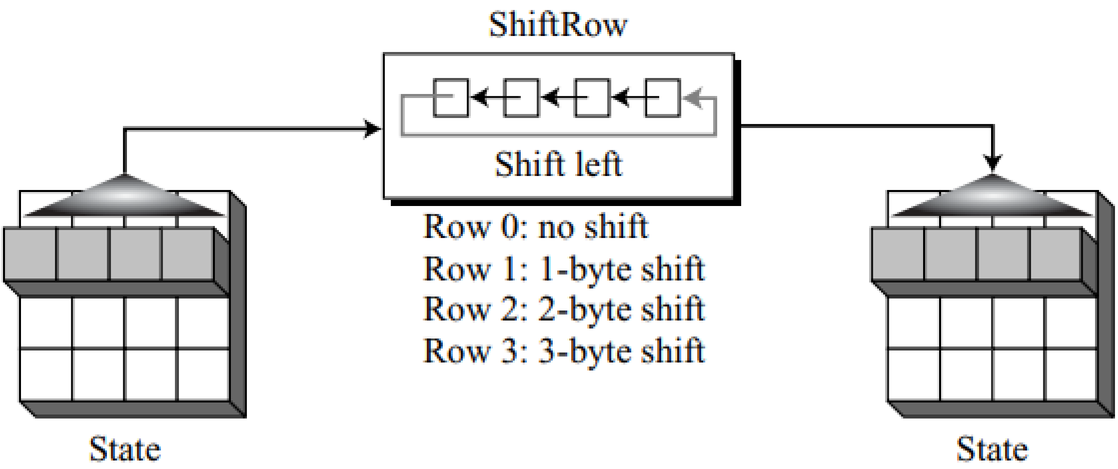
The first transformation, SubBytes, is used at the encryption site. To substitute a byte, we interpret the byte as two hexadecimal digits. The left digit defines the row and the right digit defines the column of the substitution table. The two hexadecimal digits at the junction of the row and the column are the new byte.

The SubBytes operation involves 16 independent byte-to-byte transformations.



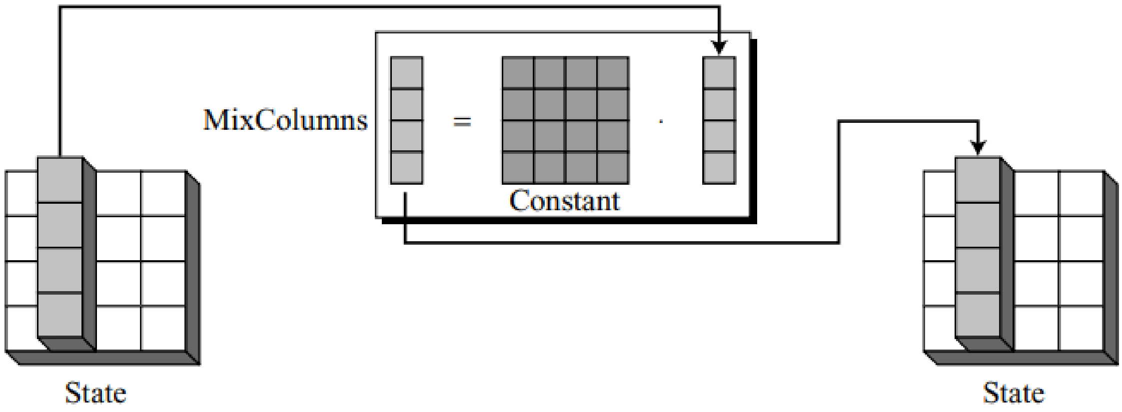
ShiftRows

In the encryption, the transformation is called ShiftRows and the shifting is to the left.The number of shifts depends on the row number (0, 1, 2, or 3) of the state matrix. This means the row 0 is not shifted at all and the last row is shifted three bytes.



MixColumns

The MixColumns transformation operates at the column level; it transforms each column
of the state to a new column. The transformation is actually the matrix multiplication of a
state column by a constant square matrix



AddRoundKey

AddRoundKey also proceeds one column at a time. The AddRoundKey transformation can be thought as XORing of each column of the state, with the corresponding key word.